THE ULTIMATE PHONE BOOK

Android Project Documentation

AREHMAN JAMAL
IYER RAHUL RAJAN
ZUBIN KADVA

Table of Contents

Role of Individual Members

Member Name	Role
Arehman Jamal	Design, Consultancy, Programming, Testing, Deployment
Iyer Rahul Rajan	Design, Code analysis, Database analysis, Query analysis
Zubin Kadva	Design, Code analysis, Programming, Testing, Code synthesis, Optimization

Project Description

Project Overview

The main emphasis of this project entitled – "**The Ultimate Phone Book**" is to provide a simple and basic but powerful phone and contact directory.

Apart from using the contact directory (Contacts) by the **Android** ® operating system, this project makes use of its own personal and private storage.

There are many forms of storage including private variables, files and even databases.

Out of these storage systems, this phone book makes use of databases viz. SQLite. Refer to the requirements section to learn about this projects' requirements.

This system begins with quite an interesting splash screen, followed by a marvel of options including:

- ✓ Add
- ✓ View
- ✓ Search (Wildcard-based)
- ✓ Update
- ✓ Filter
    - City
    - State
    - Country

The Purpose of the Project

The main objectives of this project include:

- An interesting and attractive splash screen.
- To provide users with an easy-to-use interface with self-explanatory menu icons and buttons with easy navigation.
- Ability to add an “avatar” – a small image identifying people.
- Even the ability to have a “default” avatar.
- Easy listing of all available contacts.
- Full validations with “groovy” error messages.

Following assumptions have been made:

- The user wants to stay away from the automatically syncing Contacts system.
- The user wants to keep his contacts safe and secure.
- City, State, Country must be manually entered by the user.

Requirements

Functional Requirements

- The “Add” activity requires that all fields need to be filled ex. Phone 2 and Email address.
- Strict checks for incorrect or invalid email address.
- The phone numbers field accepts only numeric values (0-9).

Data Requirements

- An automatic “age” function calculating the contact age given the users’ date of birth.
- An interactive DatePicker dialog for selecting day, month, year.
- The option to revert changes made to date of birth with an optional “Cancel”.

Security Requirements

The system itself provides some security mechanisms. But there are some which are left undiscovered. These are listed below:

- Database encryption techniques to further strengthen the security aspect.
- Secure communication between the application and its SD card storage so as to prevent other harmful apps from intercepting the sent data.
- A user can add, update the contact, but will not be able to delete an existing one.

Design

System Design

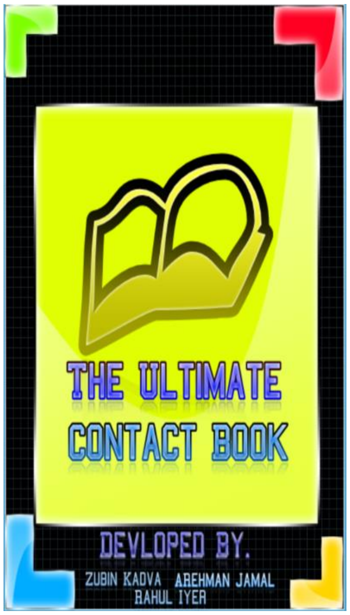
The SQLite database selected consists of only one table viz. “contacts” with the following structure:

Column Name	Data Type	Constraints
id	numeric	not null
first_name	varchar	
middle_name		
last_name		
phone1	numeric	
phone2		-
email	varchar	
dob	date	not null
age	numeric	
gender	char	
address	varchar	
city		
state		
country		
family_members	numeric	
avatar	varchar	

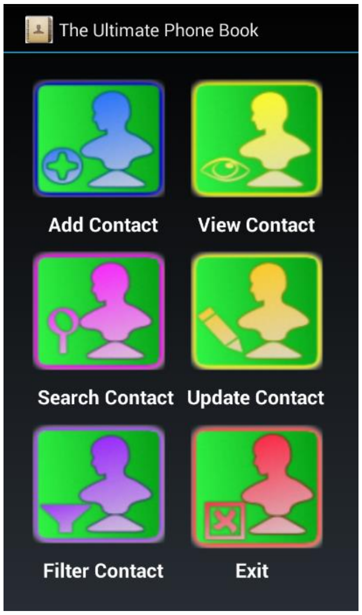
Minimum API Level 15 – **Android 4.0.3** (Ice Cream Sandwich)

This app is optimized for API Level 19 – **Android 4.4** (KitKat)

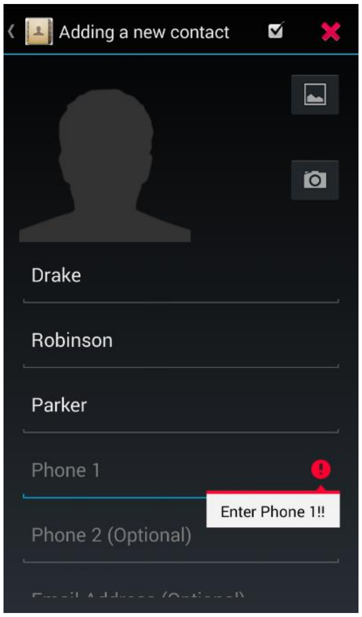
User Interface



The Splash



The Home


Adding a new contact

Drake

Robinson

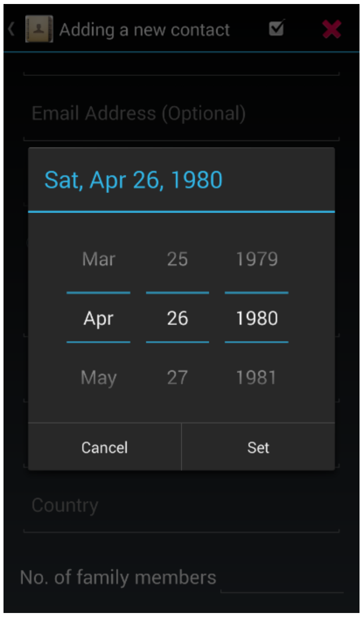
Parker

Phone 1

Enter Phone 1!!

Phone 2 (Optional)

The Add


Adding a new contact

Email Address (Optional)

Sat, Apr 26, 1980

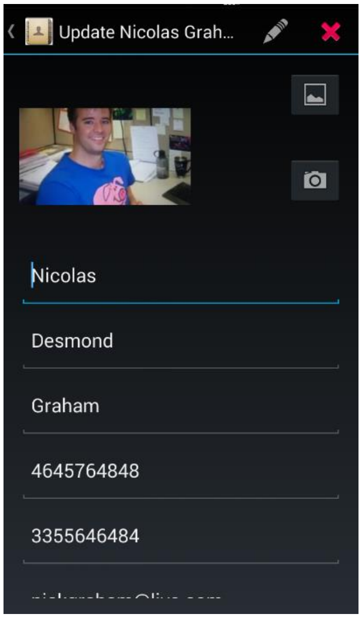
Mar	25	1979
Apr	26	1980
May	27	1981

Cancel Set

Country

No. of family members

The DatePicker


Update Nicolas Grah...

Nicolas

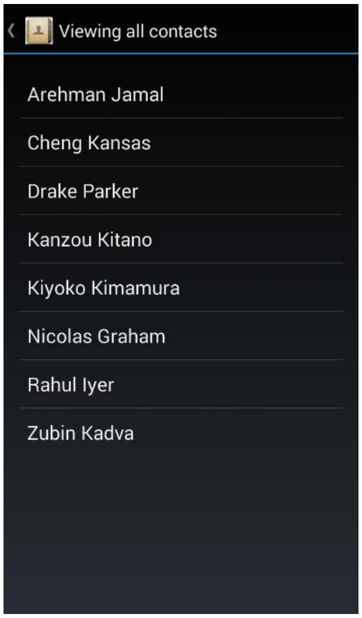
Desmond

Graham

4645764848

3355646484

The Update


Viewing all contacts

Arehman Jamal

Cheng Kansas

Drake Parker

Kanzou Kitano

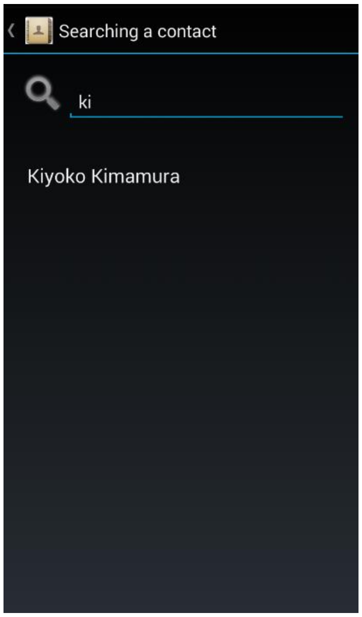
Kiyoko Kimamura

Nicolas Graham

Rahul Iyer

Zubin Kadva

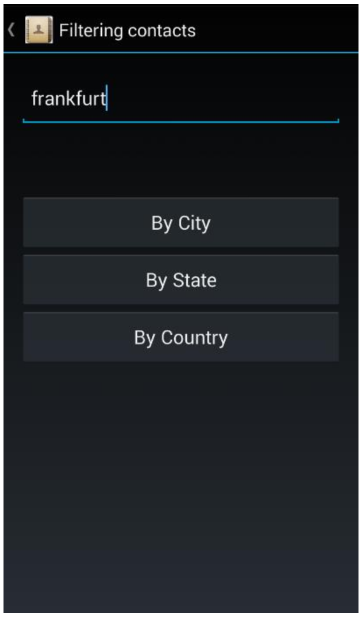
The View


Searching a contact

ki

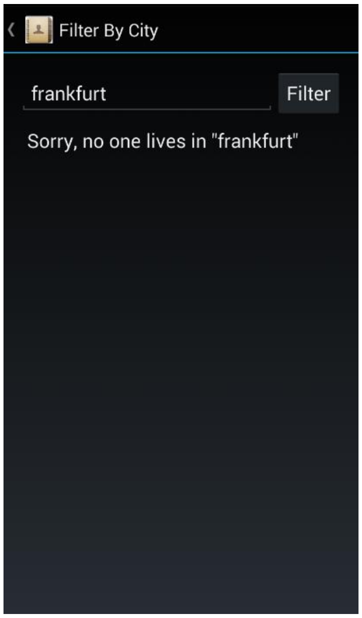
Kiyoko Kimamura

The Search


Filtering contacts

frankfurt

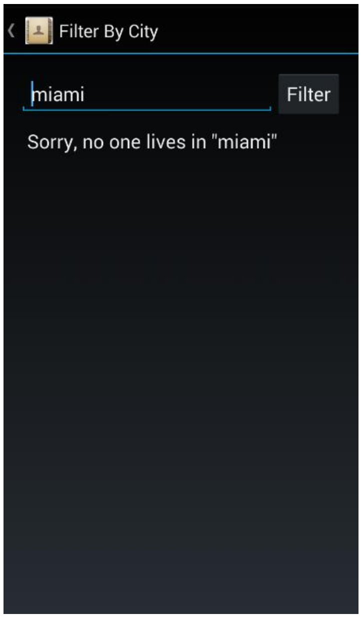
The Filter



frankfurt

Sorry, no one lives in "frankfurt"

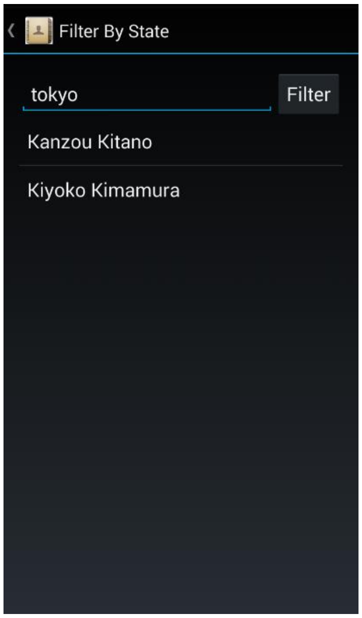
The Filter Unsuccessful



miami

Sorry, no one lives in "miami"

The Filter By City

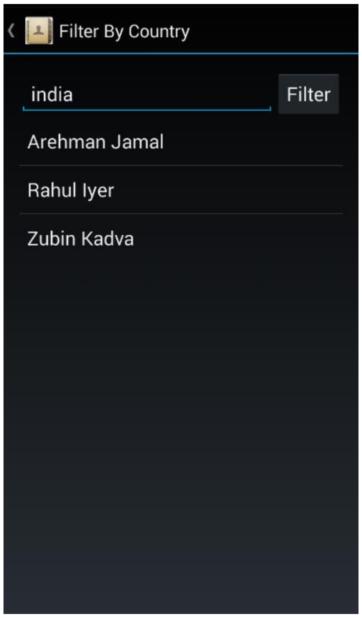


tokyo

Kanzou Kitano

Kiyoko Kimamura

The Filter By State



india

Arehman Jamal

Rahul Iyer

Zubin Kadva

The Filter By Country

Code Snippets

Splash.java

```
private static int SPLASH_TIME_OUT = 3000;

new Handler().postDelayed(new Runnable() {
          @Override
          public void run() {
             // This method will be executed once the timer is over

             Intent i = new Intent(Splash.this, Home.class);
             startActivity(i);

             // close this activity
             finish();
          }
       }, SPLASH_TIME_OUT);
```

Home.java

```
protected void createDB() {
        mydatabase = openOrCreateDatabase("zarphonebook",MODE_PRIVATE,null);
        mydatabase.execSQL("create table if not exists contacts (" +
                "id numeric," +
                "first_name varchar," +
                "middle_name varchar," +
                "last_name varchar," +
                "phone1 varchar," +
                "phone2 varchar," +
                "email varchar," +
                "dob date," +
                "age numeric, " +
                "gender char," +
                "address varchar," +
                "city varchar," +
                "state varchar," +
                "country varchar," +
                "family_members numeric," +
                "avatar varchar" +
                ");");
        mydatabase.close();
        }

createDB();
```

```
        //Menu button listeners begin here
        Button add=(Button)findViewById(R.id.cmdAdd);
        add.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                intent=new Intent(getApplicationContext(),AddUpdateContact.class);
                startActivityForResult(intent,REQUEST_CODE);
            }
        });
```

<Similar code for View, Update, Search, Filter>

```
        //Check if SD card mounted
        if
(!Environment.getExternalStorageState().equals(Environment.MEDIA_MOUNTED))
{
            Log.d("MyApp", "No SDCARD");
        }
        //SD card mounted, now create images directory
        else {
            File directory = new
File("/data/data/com.zar.theultimatephonebook/"+"images");
            directory.mkdirs();
        }
```

ViewContacts.java

```
//Check if user has come from the Update contact activity
        if(getIntent().getBooleanExtra("ToUpdateContact", false))
            setTitle("Select a contact");

        //List and ArrayAdapter creation and initialization
        list = new ArrayList<String>();
        l1 = (ListView)findViewById(R.id.lstContacts);
        ada = new ArrayAdapter(this,android.R.layout.simple_list_item_1,list);

        mydatabase = openOrCreateDatabase("zarphonebook",MODE_PRIVATE,null);
        resultSet = mydatabase.rawQuery("Select first_name || ' ' || last_name from contacts order by lower(first_name)",null);
        resultSet.moveToFirst();

        if(resultSet.moveToFirst()) {
            while (!resultSet.isAfterLast()) {
                list.add(resultSet.getString(0));
                resultSet.moveToNext();
            }
        }
        else
            list.add("No Contacts Exist Yet!!");
```

```
    l1.setAdapter(ada);
    mydatabase.close();


    //User selects a contact to view / update
    l1.setOnItemClickListener(new AdapterView.OnItemClickListener() {
      @Override
      public void onItemClick(AdapterView<?> adapterView, View view, int i,
long l) {


        //If user has selected for updating a contact
        if(getIntent().getBooleanExtra("ToUpdateContact", false))
          intent=new Intent(getApplicationContext(),AddUpdateContact.class);


        //If user has selected for viewing a contact
        else
          intent=new Intent(getApplicationContext(),ContactDetails.class);


        intent.putExtra("SelectedName",l1.getItemAtPosition(i).toString());
        startActivityForResult(intent,REQUEST_CODE);
        finish();
      }      });
```

ContactDetails.java

```
//List and ArrayAdapter creation and initialization
        list = new ArrayList<String>();
        l1 = (ListView)findViewById(R.id.lstDetails);
        ada = new ArrayAdapter(this,android.R.layout.simple_list_item_1,list);

        //ImageView definition
        avatar=(ImageView)findViewById(R.id.avatar);

        //Get name of contact passed as an intent
        gotName=getIntent().getStringExtra("SelectedName");
        setTitle(gotName);



        //Retrieve contact id depending on parameter passed
        mydatabase = openOrCreateDatabase("zarphonebook",MODE_PRIVATE,null);
        resultSet = mydatabase.rawQuery("select id from contacts where first_name
|| ' ' || last_name='"+gotName+"' ",null);
        resultSet.moveToFirst();

        while(!resultSet.isAfterLast()) {
            gotId=resultSet.getInt(0);
            resultSet.moveToNext();
        }
```

```
    //Now retrieve full contact details
    resultSet = mydatabase.rawQuery("select * from contacts where
id="+gotId+" ",null);
    resultSet.moveToFirst();

    while(!resultSet.isAfterLast()) {
      //Index 1 - Full name (First / Middle / Last name) …
      //Index 2 - Phone 1 …
      //Index 3 - Phone 2, but first check if null or what …
      //Index 4 - Email, but first check if null or what …
      //Index 5 - DOB, age, gender …
      //Index 6 - Full address …
      //Index 7 - City, state, country …
      //Index 8 - Number of family members …
      //Time for the avatar …
    //Retrievals complete

    l1.setAdapter(ada);
    mydatabase.close();

    //User selects an item from the list, hence add a listener
        //If user has selected Phone 1 call
        if(l1.getItemIdAtPosition(i)==1) {
          intent = new Intent(Intent.ACTION_CALL);
          intent.setData(Uri.parse("tel:"+tel1));
          startActivity(intent);
        }
```

```
                //If user has selected Phone 2 call, but check if number is present first!!
                else if(l1.getItemIdAtPosition(i)==2 &&
!l1.getItemAtPosition(i).equals("<Phone 2 Absent>")) {
                    intent = new Intent(Intent.ACTION_CALL);
                    intent.setData(Uri.parse("tel:"+tel2));
                    startActivity(intent);
                }

                //If user has selected email, but check if address is present first!!
                else if(l1.getItemIdAtPosition(i)==3 &&
!l1.getItemAtPosition(i).equals("<Email Address Absent>")) {
                    intent = new Intent(Intent.ACTION_SEND);
                    intent.setType("message/rfc822");
                    intent.putExtra(Intent.EXTRA_EMAIL  , new String[]{eadd});

                    try {
                        startActivity(intent);
                    }
                    catch (android.content.ActivityNotFoundException ex) {
                        Toast.makeText(ContactDetails.this, "There are no email clients
installed.", Toast.LENGTH_SHORT).show();
                    }
                }
            }
        });
```

AddUpdateContact.java

```
        //Control definitions …
        //Setting the file name for a newly captured image
        String name = dateToString(new Date(),"yyyy-MM-dd-hh-mm-ss");
        destination = new File(Environment.getExternalStorageDirectory(), name +
".jpg");

        //If user wants to select an image from the gallery
        gallery.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent();
                intent.setType("image/*");
                intent.setAction(Intent.ACTION_GET_CONTENT);
                startActivityForResult(Intent.createChooser(intent,"Select Picture"),
SELECT_PICTURE);
            }        });

        //If user wants to capture an image
        camera.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                intent.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(destination));
                startActivityForResult(intent, REQUEST_IMAGE);
            }        });
```

```
    //Show the date picker …


    //Display default male image when page is first loaded (Male is by default checked)

    display.setBackgroundResource(R.drawable.default_male);


    //If user explicitly clicks on Male

    male.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {

        @Override

        public void onCheckedChanged(CompoundButton compoundButton, boolean b) {

            if(imagePath==null || imagePath.equals("null"))

                display.setBackgroundResource(R.drawable.default_female);

        }       });


    //If user explicitly clicks on Female

    female.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {

        @Override

        public void onCheckedChanged(CompoundButton compoundButton, boolean b) {

            if(imagePath==null || imagePath.equals("null"))

                display.setBackgroundResource(R.drawable.default_male);

        }

    });
    //If the request has come from Update activity …
        //First get the ID …
        //Then retrieve the contact details based on the above ID …
            //Part 1 - Full name (First / Middle / Last name) …
            //Part 2 - Phones and email …
            //Part 3 – DOB …
            //Part 4 – Gender …
            //Part 5 - Full address …
            //Part 6 - City, state, country …
            //Part 7 - Number of family members …
            //Part 8 - Avatar [The most sickening part!!] …
//If the avatar has no value, then display default (Of course, based on gender!!)
            if(imagePath==null || imagePath.equals("null")) {
      if (resultSet.getString(resultSet.getColumnIndex("gender")).equals("M"))
                    display.setBackgroundResource(R.drawable.default_male);
                else
            display.setBackgroundResource(R.drawable.default_female);            }
            //Avatar has a value
            else {
                Bitmap bitmap = BitmapFactory.decodeFile(imagePath);
                display.setBackgroundResource(0);
                display.setImageBitmap(bitmap);              }
            resultSet.moveToNext();           }
        //Retrieval done!!
        mydatabase.close();
    }
}
```

SearchContact.java

```
ImageButton b1;
EditText search;
String query;
ArrayList<String> list;
ListView l1;
ArrayAdapter ada;
SQLiteDatabase mydatabase;
Cursor resultSet;
Intent intent;
final int REQUEST_CODE=10;

//List and ArrayAdapter creation and initialization
        list = new ArrayList<String>();
        l1 = (ListView)findViewById(R.id.lstResults);
        ada = new ArrayAdapter(this,android.R.layout.simple_list_item_1,list);

        //Control definitions
        b1 = (ImageButton)findViewById(R.id.cmdGo);
        search = (EditText)findViewById(R.id.txtQuery);

        //User clicks on Search button
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(search.getText().length()==0) {
```

```
          search.setError("Enter Search Criteria!!");          }
        else {                list.clear();
   mydatabase = openOrCreateDatabase("zarphonebook", MODE_PRIVATE, null);
          query = search.getText().toString().toLowerCase();
          resultSet = mydatabase.rawQuery("select first_name || ' ' ||
last_name from contacts where lower(first_name) like '" + query + "%'", null);
          resultSet.moveToFirst();
          if (resultSet.moveToFirst()) {
            while (!resultSet.isAfterLast()) {
              list.add(resultSet.getString(0));
              resultSet.moveToNext();             }
          } else
            list.add("Sorry, no one named \"" + search.getText() + "\"");


          mydatabase.close();
          l1.setAdapter(ada);        }      }     });


    //User selects a contact to view
    l1.setOnItemClickListener(new AdapterView.OnItemClickListener() {
      @Override
      public void onItemClick(AdapterView<?> adapterView, View view, int i,
long l) {
        intent=new Intent(getApplicationContext(),ContactDetails.class);
        intent.putExtra("SelectedName",l1.getItemAtPosition(i).toString());
        startActivityForResult(intent,REQUEST_CODE);
      }
    });
```

FilterContacts.java

```
final int REQUEST_CODE=10;
Intent intent;
EditText filter;

//Filter criteria text box
        filter=(EditText)findViewById(R.id.txtFilter);

        //Filter by city button click listener
        Button city=(Button)findViewById(R.id.cmdCity);
        city.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(filter.getText().length()==0)
                    filter.setError("Enter Filter Criteria!!");
                else {
                    intent = new Intent(getApplicationContext(), FilterCity.class);
                    intent.putExtra("Filter", filter.getText().toString().toLowerCase());
                    startActivityForResult(intent, REQUEST_CODE);
                }
            }
        });

        //Filter by state button click listener
        Button state=(Button)findViewById(R.id.cmdState);
        state.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
    public void onClick(View view) {
        if(filter.getText().length()==0)
            filter.setError("Enter Filter Criteria!!");
        else {
            intent = new Intent(getApplicationContext(), FilterState.class);
            intent.putExtra("Filter", filter.getText().toString().toLowerCase());
            startActivityForResult(intent, REQUEST_CODE);
        }
    }
});

//Filter by country button click listener
Button country=(Button)findViewById(R.id.cmdCountry);
country.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(filter.getText().length()==0)
            filter.setError("Enter Filter Criteria!!");
        else {
            intent = new Intent(getApplicationContext(), FilterCountry.class);
            intent.putExtra("Filter", filter.getText().toString().toLowerCase());
            startActivityForResult(intent, REQUEST_CODE);
        }
    }
});
```

FilterCity

```
        //Filter button click listener …


    //Logic to filter by city
    protected  void filterByCity() {
        list.clear();
        mydatabase = openOrCreateDatabase("zarphonebook",MODE_PRIVATE,null);
        resultSet = mydatabase.rawQuery("Select first_name || ' ' || last_name from
contacts where lower(city)='"+criteria+"'order by lower(first_name)",null);
        resultSet.moveToFirst();

        if(resultSet.moveToFirst()) {
            while (!resultSet.isAfterLast()) {
                list.add(resultSet.getString(0));
                resultSet.moveToNext();
            }
        }
        else {
            list.add("Sorry, no one lives in \""+criteria+"\"");
        }

        l1.setAdapter(ada);
        mydatabase.close();
}
```

FilterState.java

```
        //Filter button click listener …


    //Logic to filter by state
    protected  void filterByState() {
        list.clear();
        mydatabase = openOrCreateDatabase("zarphonebook",MODE_PRIVATE,null);
        resultSet = mydatabase.rawQuery("Select first_name || ' ' || last_name from
contacts where lower(state)='"+criteria+"'order by lower(first_name)",null);
        resultSet.moveToFirst();

        if(resultSet.moveToFirst()) {
            while (!resultSet.isAfterLast()) {
                list.add(resultSet.getString(0));
                resultSet.moveToNext();
            }
        }
        else {
            list.add("Sorry, no one lives in \""+criteria+"\"");
        }

        l1.setAdapter(ada);
        mydatabase.close();
    }
```

FilterCountry

```
        //Filter button click listener …


    //Logic to filter by country
    protected  void filterByCountry() {
        list.clear();
        mydatabase = openOrCreateDatabase("zarphonebook",MODE_PRIVATE,null);
        resultSet = mydatabase.rawQuery("Select first_name || ' ' || last_name from
contacts where lower(country)='"+criteria+"'order by lower(first_name)",null);
        resultSet.moveToFirst();

        if(resultSet.moveToFirst()) {
            while (!resultSet.isAfterLast()) {
                list.add(resultSet.getString(0));
                resultSet.moveToNext();
            }
        }
        else {
            list.add("Sorry, no one lives in \""+criteria+"\"");
        }

        l1.setAdapter(ada);
        mydatabase.close();
    }
```

References and Bibliography

http://developer.android.com/

http://stackoverflow.com/

http://coronalabs.com/

http://www.appcelerator.com/